NIST Special Publication 800-106
Draft
Randomized Hashing Digital Signatures

Quynh Dang

Computer Security Division
Information Technology Laboratory

C O M P U T E R S E C U R I T Y

July 2007

U.S. Department of Commerce
Carlos M. Gutierrez, Secretary
Technology Administration
Robert Cresanti, Under Secretary of Commerce for Technology
National Institute of Standards and Technology
William Jeffrey, Director

NIST National Institute of Standards and Technology • Technology Administration • U.S. Department of Commerce

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of non-national security-related information in Federal information systems. This special publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

A digital signature is generated by using a hash function and a digital signature algorithm. Approved hash functions and digital signature algorithms can be found in [FIPS 180-3] and [FIPS 186-3], respectively. The security provided by the hash functions is vital to the security that can be provided by digital signatures. This Recommendation specifies a method to strengthen the security of the hash functions in digital signature applications.

KEY WORDS: digital signature, hash function, collision resistance, randomized hashing.

Acknowledgements

The author, Quynh Dang of the National Institute of Standards and Technology (NIST) gratefully acknowledges and appreciates contributions by Hugo Krawczyk, Elaine Barker, John Kelsey, W. Timothy Polk, Donna F. Dodson, Shu-jen Chang and William E. Burr concerning the many security and applicability issues associated with this Recommendation.

Table of Contents

1. Introduction

This recommendation provides a technique to randomize the input messages to hash functions prior to the generation of digital signatures using the Digital Signature Algorithm (DSA), Elliptic Curve Digital Signature Algorithm (ECDSA) and RSA. Approved hash functions for Federal government use are specified in Federal Information Processing Standard (FIPS) 180-3, the Secure Hash Standard (SHS) [FIPS 180-3]. Digital Signatures shall be generated as specified in FIPS 186-3, the Digital Signature Standard [FIPS 186-3].

Collision resistance is a required property for the hash functions used in Digital Signature Applications. The intent of this randomization technique is to strengthen the collision resistance provided by the hash functions in digital signature applications without any changes to the core hash functions and digital signature algorithms. A message will have a different digital signature each time it is signed if it is randomized with a different random value. This Recommendation is based on the work of Shai Halevi and Hugo Krawczyk [Randomizing].

1.1 Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This recommendation is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This Recommendation has been prepared for use by Federal agencies. It may be used by non-governmental organizations on a voluntary basis and is not subject to copyright. Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should this Recommendation be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of OMB, or any other Federal official.

1.2 Audience and Assumptions

This Recommendation is targeted at Federal agencies and implementers of digital signature applications. Readers are assumed to have a working knowledge of cryptography, especially the cryptographic hashing algorithms specified in [FIPS 180-3] and their use by the digital signature algorithms specified in [FIPS 186-3].

1.3 Glossary

Approved	FIPS-Approved and/or NIST-recommended. An algorithm or technique that is either 1) specified in a FIPS or NIST Recommendation, or 2) adopted in a FIPS or NIST Recommendation and specified either in an appendix to the FIPS or NIST Recommendation.
Collision	Two different messages have the same hash value from the same hash function.
Byte	Eight binary bits.
Collision resistance	The number of executions of the hash function that would be needed to find two messages with the same hash value.
Digital signature	The result of applying some cryptographic functions to data that, when the functions are properly implemented, provides origin authentication, data integrity and signatory non-repudiation.
Entropy	A measure of the disorder, randomness (See Random bit) or variability in a closed system. The entropy of X is a mathematical measure of the amount of information provided by an observation of X. As such, entropy is always relative to an observer and his or her knowledge prior to an observation.
Hash function	A function that maps a bit string of arbitrary length to a fixed length bit string. Approved hash functions are specified in FIPS 180-3 and satisfy the following properties: 1. (One-way) It is computationally infeasible to find any input that maps to any pre-specified output, and 2. (Collision Resistant) It is computationally infeasible to find any two distinct inputs that map to the same output.
Hashed	Data has been processed by a hash function (see hash function) to produce a hash value (see hash value).
Hashing algorithm	A set of steps to execute a hash function. See “hash function”.
Hash value	The result of applying a hash function to a message. Also known as a “message digest” or a “hash output”.
Random bit	A binary bit which an attacker has only 50% probability of success of guessing the value of the bit either zero or one.
Random value	Random bits. (See Random bit)

Randomized hashing	A technique for which input to a hash function is randomized before it is hashed.
Randomized message	A message that has been modified using a random value.
Salt	A random value generated during digital signature generation using the *RSA Signature Scheme with Appendix - Probabilistic Signature Scheme (RSASSA-PSS RSA)* [PKCS#1 v2.1].
Second preimage	Given two different messages *A* and *B* with hash values *hash*(*A*) and *hash*(*B*), respectively, if *hash*(*A*) = *hash*(*B*), then *B* is a second preimage of *hash*(*A*).
Second preimage resistance	It is computationally infeasible to find a second input that has the same hash value as any other specified input. That is, given an input *A*, it is computationally infeasible to find a second input *B* that is different from *A*, such that *hash*(*A*) = *hash* (*B'*).
Sig	A digital signature of a randomized message.

1.4 Abbreviations and Terms

DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
PKCS	Public Key Cryptography Standard
RSA	Rivest-Shamir-Adleman
RSASSA-PSS	RSA Signature Scheme with Appendix - Probabilistic Signature Scheme (RSASSA-PSS) as specified in Public Key Cryptography Standard (PKCS) #1 [PKCS#1 v2.1].
SP	Special Publication

1.5 Symbols

M	The randomized message.
Ms	The (original) message that is signed.
padding	Either a string of binary bit zero(s) or the empty string.

padding_length	A 16-bit string that indicates the length of *padding*.
rv	The random value.
(r, s)	Digital signature for DSA and ECDSA.
$\|x\|$	Length of x in bits.
$+$	Addition. For example, $5 + 4 = 9$
$\oplus$	Bitwise logical "exclusive-or", where $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, and $1 \oplus 1 = 0$. Example: $01101 \oplus 11010 = 10111$
$\lfloor a \rfloor$	The floor of a non-negative number a: the greatest integer that is smaller than or equal to a. For example, $\lfloor 5 \rfloor = 5$, and $\lfloor 5.9 \rfloor = 5$
$\|\|$	Concatenation; e.g. $A \| \| B$ is the concatenation of A and B.
0^x	A string of x zero bits; e.g., $0^5 = 00000$.

2. Rationale for Using Randomized Hashing for Digital Signatures

The randomizing technique specified later in this Recommendation is applicable to new and existing digital signature applications which require collision resistance hash functions. In the original schemes for generating digital signatures specified in [FIPS 186-3] (i.e., without randomized hashing), the attacker may be able to calculate the hash value of the message that he/she wants to have signed by the signer; a possible attack scenario is the following:

> The attacker calculates hashed values for messages until a pair of messages are found to have the same hash value. At least one of the messages (message *A*) must be chosen such that a signer would be willing to sign it; the other message (message *B*) would be chosen such that its contents would be beneficial to the attacker in some manner. The attacker asks the signer to sign message *A*, but the attacker uses the signature computed on message *A* for the other message (message *B*) that the signer did not sign. This attack is computationally possible if the hash function is not strongly collision resistant.

When using the randomized hashing technique described for the digital signatures schemes in this Recommendation, the attacker does not know what the hash value of the randomized message will be before the digital signature is generated. Therefore, the attacker cannot conduct the collision attack described above. However, the attacker may be able to conduct other attacks on the hash functions. This Recommendation is only aimed at strengthening the collision resistance of the hash values generated during digital signature generation. In particular, in order to forge signatures that use the randomized hashing technique, the attacker needs to find second

preimages of the hash values, defeating the second preimage resistance of the hash function, a much harder task than finding collisions.

Even though the main purpose of this Recommendation is to specify the randomized hashing technique used to strengthen security of digital signatures, the technique may be used for other cryptographic applications using hash functions. This randomized hashing technique is useful only when the data to be hashed includes data provided by another party different from the party who randomizes and then hashes the data. See Figure 1 below.

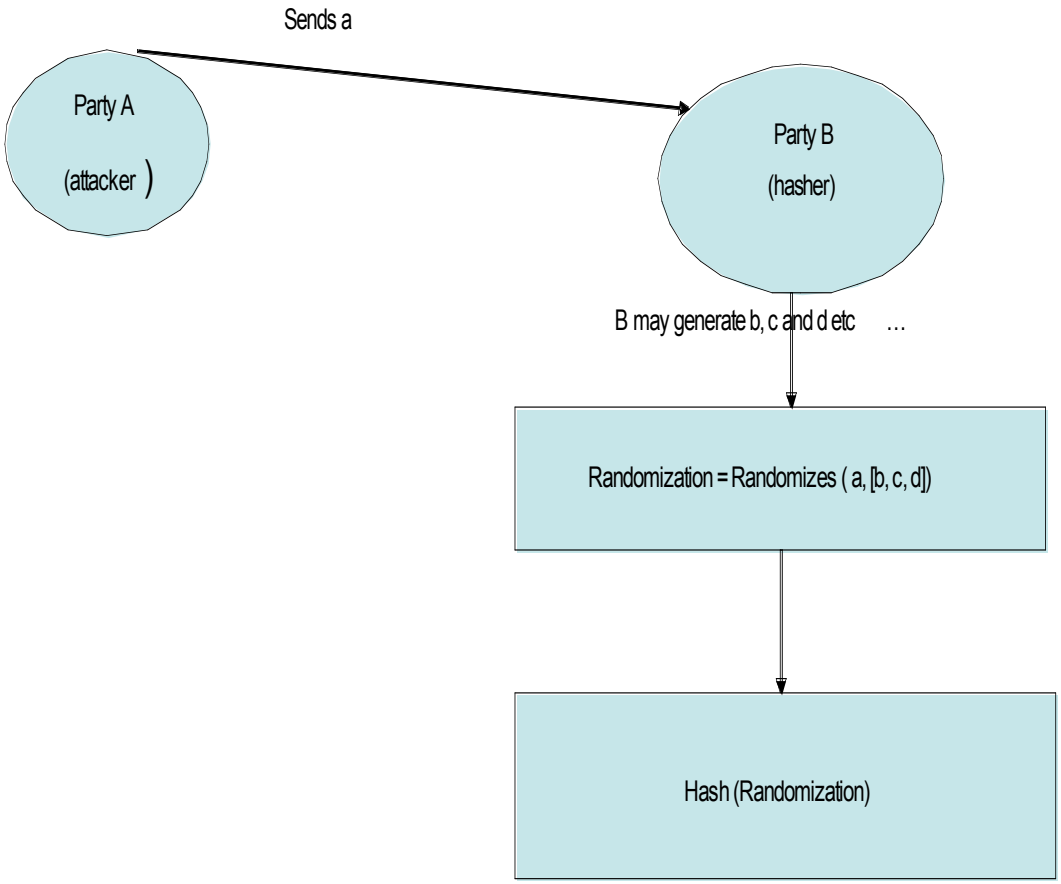

Figure 1: Randomization scenario

If the data party B randomizes and hashes contains only data he/she generates (b, c, d etc…in the Figure 1), then the randomization technique does not provide any additional security other than collision resistance of the hash function. The randomization technique is useful as long as the data party B randomizes and hashes contains data (such as a in the Figure 1) generated by another party (party A).

3. Randomized Hashing

3.1 Randomizing a Message Prior to Hashing

To generate a digital signature for a message, the message is hashed by using one of the Approved hashing algorithms [FIPS 180-3] first, and then signing the resulting hash value using one of the Approved digital signature algorithms [FIPS 186-3]. The technique specified in this Recommendation results in a transformation of a message before providing the (transformed) message to the hash function. The goal of the randomizing technique specified below is to be independent of the hash functions used in all digital signature schemes. Therefore, one implementation of the technique can be used for all digital signature applications. The following are required operations to randomize a message.

3.2 Message Randomization

A random value *rv* must be obtained to randomize a message. To utilize all the entropy in the random value, the entire random value string must be used. Any method used to reduce the length of the random value string (e.g., by hashing) may result in a loss of entropy; the entropy in the resulting string will be no more than the length of that string. For example, if the random value is 256 bits long and contains 256 bits of entropy, when it is processed to become a 128-bit long string, then this new 128-bit string will contain no more than 128 bits of entropy. The randomization method in this Recommendation does not reduce the length of the random value.

The primary function used to randomize the message with the random value in this Recommendation is the bitwise logical "exclusive-or" operation, $\oplus$, which is defined in Section 1.5. To use the entire random value string without reducing its length an encoded message string is produced as follows:

encoded message string = *Ms* || *padding* || *padding length*

where *Ms* is the original message, *padding* is a string of zero bits, and *padding_length* is a string of 16 bits that indicates the length of the *padding*. When *padding* is not required (i.e., the length of *Ms* plus the length of *padding_length* is at least as long as the random value), *padding* is the empty string; in this case, *padding_length* is a string of 16 zero bits. The randomization method is described as follows:

Inputs to message randomization method:

Ms: an input message

rv: a random bit string as described in Section 3.3 below.

Output from the message randomization method:
M: a randomized message

Message Randomization (*Ms*, *rv*):
{

1. If $(16 + |Ms| \geq |rv|)$
   {
   1.1 *padding* is the empty string.

   1.2 $padding\_length = 0^{16}$.

   }

   Else
   {
   1.3 $padding = 0^{|rv| - (|Ms| + 16)}$.

   1.4 *n* is a positive integer, and $n = |padding|$ in bits.

   1.5 Convert *n* to a binary string as specified in [FIPS 186-3] to obtain *padding_length.*
   }

2. $m = Ms \,||\, padding \,||\, padding\_length$.

3. $counter = \lfloor |m| / |rv| \rfloor$.

4. $remainder = (|m| \ mod \ |rv|)$.

5. Concatenate *rv* to itself *counter* times, and then concatenate the *remainder* left-most bits of *rv* to get *rv',* such that $|rv'| = |m|$.

   $rv' = rv \,||\, rv \,||...||\, rv \,||$ (the *remainder* left-most bits of *rv*).

6 $M = rv \,||\, (m \oplus rv')$ (Figure 2).

}

Output: *M*

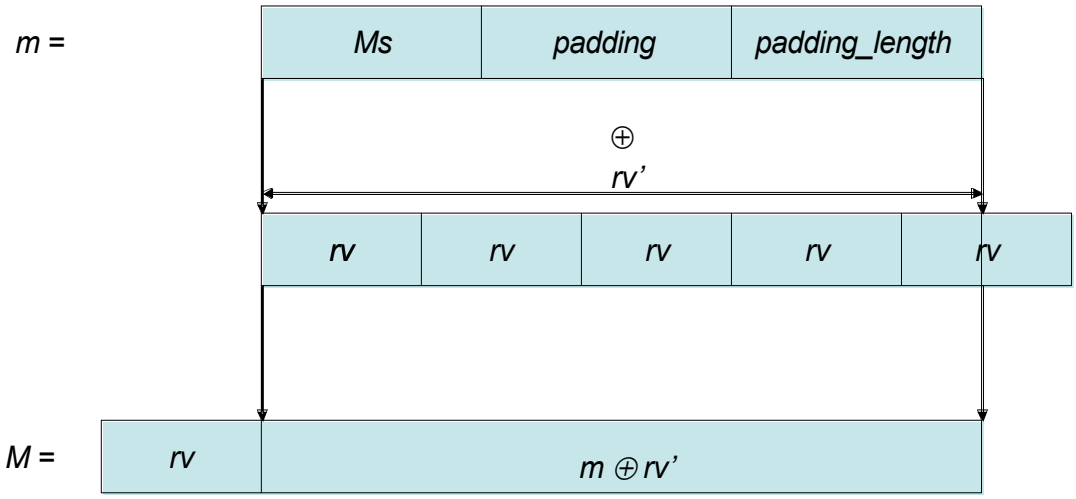

Figure 2: Message Randomization

3.3 The Random Value

The random value *rv* shall be a message-independent value that is obtained from an Approved random number generator (see [SP 800-90]) with entropy that is at least 128 bits. If the security strength *s* to be used for the digital signature scheme is known, the minimum entropy for *rv* shall be the greater of 128 and the security strength *s* (i.e., if $s < 128$, then the minimum entropy for *rv* shall be at least 128 bits; if $s \geq 128$, then the minimum entropy for *rv* shall be at least *s* bits).

rv shall be kept secret until a digital signature using this value *rv* is generated. While *rv* is kept secret, it shall be afforded the same protections as a digital signature private key (see [SP 800-57]). Note that this randomized hashing technique may not provide additional security strength over the standard hashing technique in generating digital signatures as specified in [FIPS 180-3] when the random value *rv* is known to the message generator (the attacker) before the digital signature associated with *rv* is generated.

Each *rv* shall be used for only one signature. If the same message is signed multiple times, a different value of *rv* shall be used during each signature generation process.

3.4 Hashing

The randomizing technique in Section 3.2 shall be performed before the hash function is invoked in order to produce a randomized message. This transformation requires no changes to the hash function specified in [FIPS 180-3], since the randomized message, *M*, is hashed instead of the original message, *Ms*. However, the random value *rv* must be provided to the digital signature verifier for signature verification, i.e., the signature verifier must have *rv, Ms* and the digital signature on the randomized message *M*.

4. Digital Signatures Using Randomized Hashing

Nothing is changed in the DSA, ECDSA and RSA digital signature schemes [FIPS 186-3] when using the randomized hashing technique as described in Section 3. However, to accommodate this technique, additional operations are needed during signature generation and verification, as described below:

Signature Generation:

1. Obtain the *rv* as described in Section 3.3
2. Randomize the message *M*s with *rv* as described in Section 3.2, obtaining the randomized message *M*.
3. Generate a digital signature *sig* on the randomized message *M* as specified in [FIPS 186-3].
4. Provide the original message *Ms*, the random value *rv* and the digital signature *sig* for signature verification.

Signature Verification:

1. The digital signature verifier receives the signature *sig'*, the message *Ms'* and the random value *rv'*.[1]
2. *Ms'* is randomized with *rv'* as described in Section 3.2 (*Ms'* is used as *Ms,* and *rv'* is used as *rv* in the randomization process). Let the randomized result be *M'*.
3. *M'* and the signature *sig'* are used in the signature verification and validation process as specified in [FIPS 186-3].

In DSA and ECDSA, there is an existing random value *r* that is a part of a digital signature (*r, s*), and is a message-independent value. [FIPS 186-3] allows this value to be pre-computed prior to performing a digital signature operation. When an implementation pre-computes *r*, the value of *r* may be used as the random value *rv*, though this is not required. In this case, *r* is provided twice to the signature verifier unless otherwise agreed upon, once as the *rv*, and again within the digital signature (*r*,*s*). The entropy in *r* shall meet or exceed the required entropy for *rv* if *r* is to be used as the random value *rv* (see Section 3.3).

[1] *Ms'* , *rv*' and *sig'* are purportedly *Ms, rv* and *sig*, respectively

In the RSA Signature Scheme with Appendix - Probabilistic Signature Scheme (RSASSA-PSS) digital signature scheme specified in [PKCS#1 v2.1], there is an existing random value *Salt* that is generated during digital signature generation. The *Salt* is a message-independent value. The *Salt* may be used as the random value *rv* if the following conditions are met: 1) the *Salt* is available before a message is hashed, and 2) the *Salt* satisfies the conditions for a random value as specified in Section 3.3. However, for many existing applications, one or more of these conditions are not met. Note that modifications to the existing applications may cause some compatibility issues.

5. References

[Randomizing] Shai Halevi and Hugo Krawczyk, Strengthening Digital Signatures via Randomized Hashing, Advances in Cryptology, CRYPTO 2006 Proceedings.

[FIPS 186-3] Federal Information Processing Standard 186-3, Digital Signature Standard (DSS), Draft March 2006.

[FIPS 180-3] Federal Information Processing Standard 180-3, Secure Hash Standard (SHS), Draft June 2007.

[SP 800-90] NIST Special Publication (SP) 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 06.

[SP 800-57] NIST Special Publication (SP) 800-57, Part 1, Recommendation for Key Management: General, August 2005.

[PKCS#1 v2.1] RSA Laboratories, PKCS#1 v2.1: RSA Cryptographic Standard, June 14, 2002.